Vinum Analytica:

Wine Review Insight

A Data Mining and Machine Learning Project

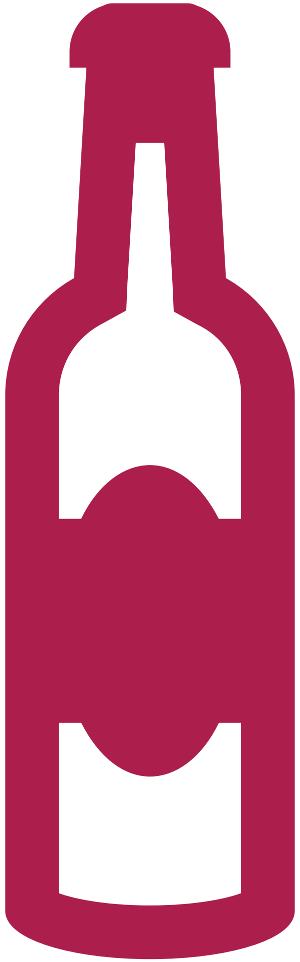
Paolo Palumbo

Introduction

Objective: Classify wines by grape variety using language from sommelier reviews.

Approach: Use machine learning models (Decision Tree, Random Forest, Neural Network).

Goal: Identify linguistic patterns that correlate with grape varieties.



Dataset Description

Source: 130k wine reviews from Kaggle.

Important Columns:

- Description, Price, Points
- Country, Region, Province
- Variety, Title, Winery

Focus: Text ***descriptions*** used to analyze sensory characteristics and classify grape ***varieties***.

Data Cleaning

Removed: 'Blend' varieties and duplicates.

Threshold: Minimum representation applied.

Final Selection: 27 distinct grape varieties for analysis.

Class Contamination

Issue: Variety names in reviews introduce bias.

Solution: Removed variety names from descriptions.

Purpose: Focus on sensory characteristics and wine descriptions.

Text Preprocessing

- Text Normalization
- Stop Words Removal
- Stemming

Goal: Clean and standardize text data for analysis.

Outcome: Improved text consistency and analysis accuracy.

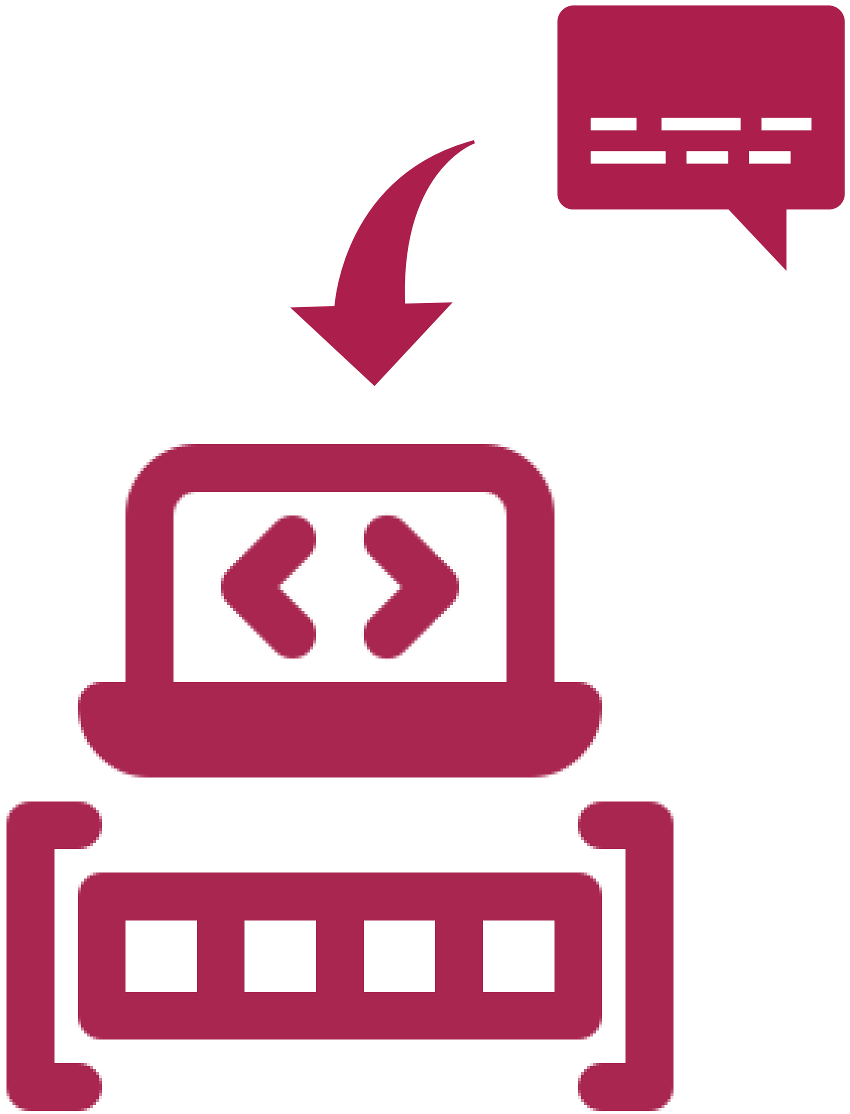
Feature Vectorization

Method: TF-IDF (Term Frequency-Inverse Document Frequency).

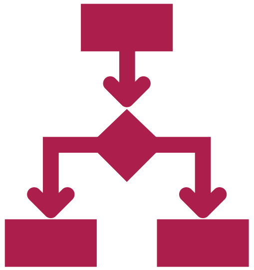
Purpose: Converts text reviews into numeric vectors.

Components:

- **Term Frequency**: Frequency of a term in a document.
- **Inverse Document Frequency**: Rarity of the term across all documents.



Model Used

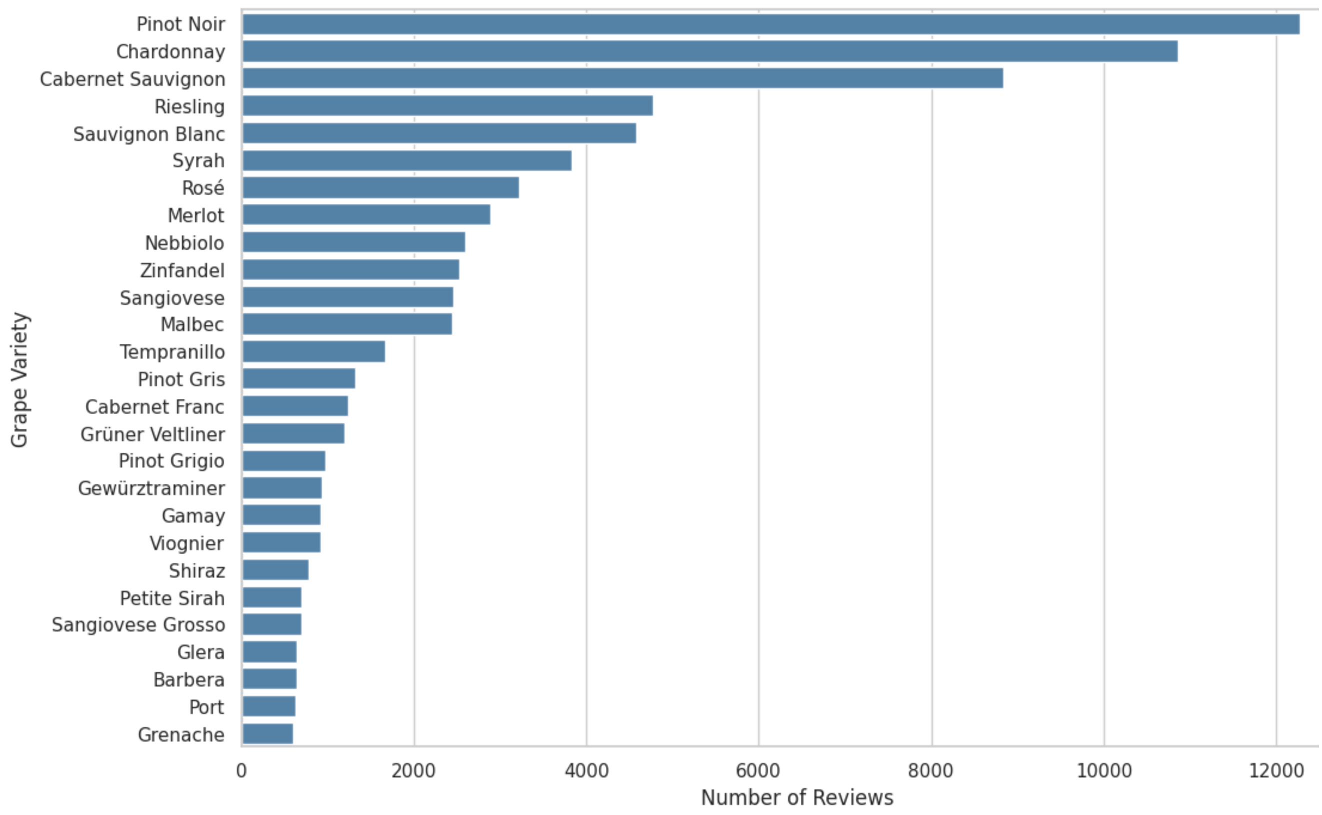


Decision Tree: Simple tree-based model for classification.

Random Forest: Ensemble of Decision Trees for improved accuracy.

Neural Network: Deep learning model for complex patterns.

Handling Class Imbalance



SMOTE: Synthetic Minority Over-sampling Technique to increase minority class samples.

Undersampling: Reduces majority class samples to balance the dataset.

Limit: Dataset size capped at 200k records.

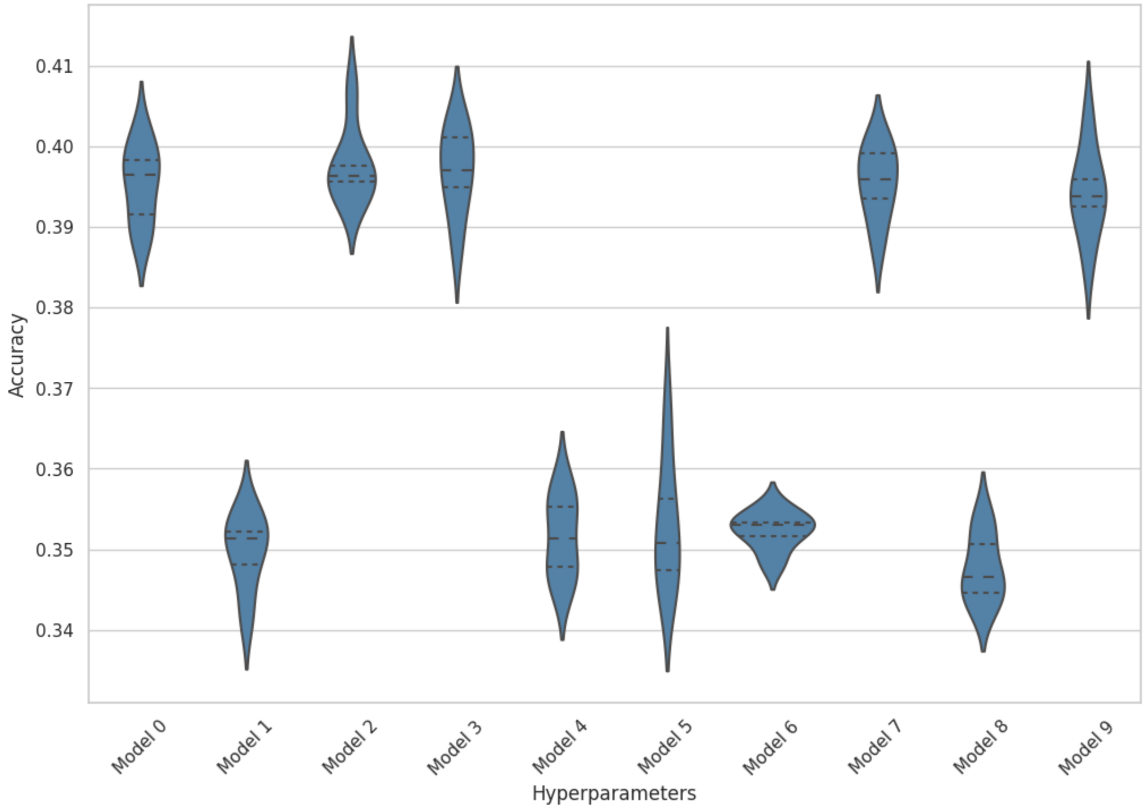
Hyperparameters Tuning

Technique: Random search with 10 combinations per model.

Validation: 6-fold cross-validation used to evaluate models.

Focus: Identified optimal hyperparameters for each model.

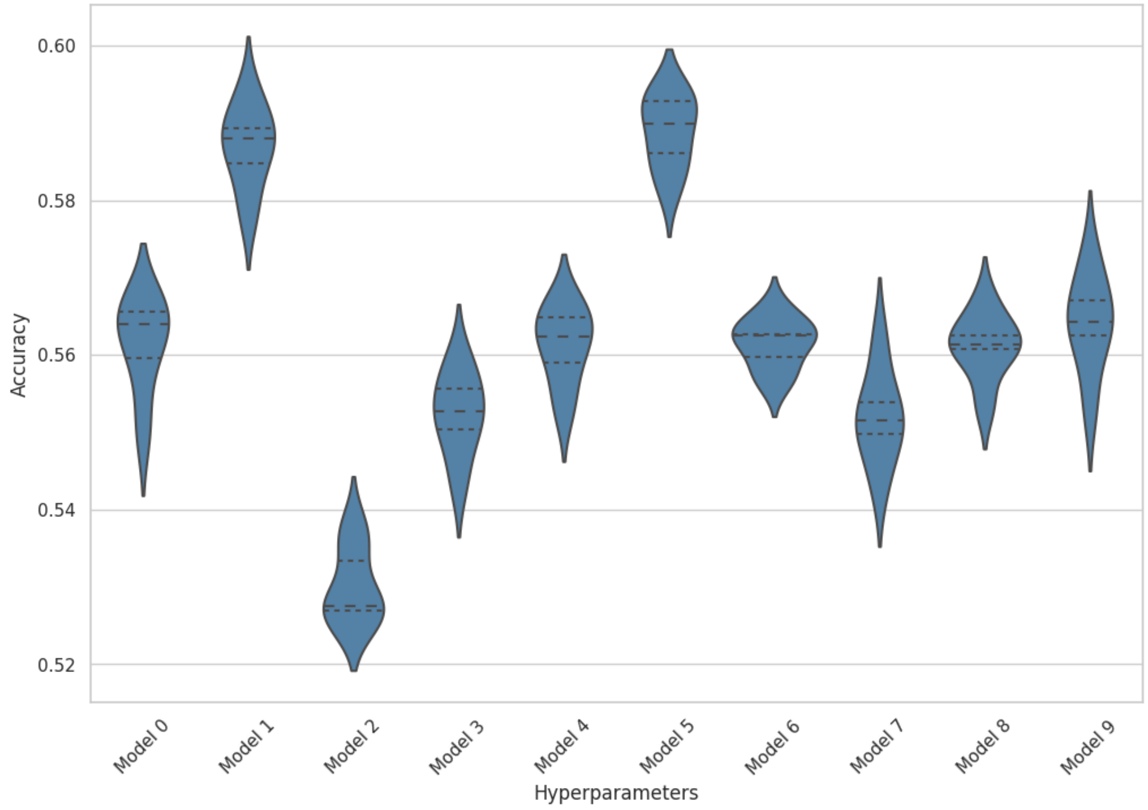
Experimental Results: Decision Tree



- **Best Model**: Gini criterion, Max Depth of 200.
- **Accuracy**: Average accuracy of 39.77%.
- **Observation**: Performance varied with hyperparameters.

Hyperparameter	Value
Criterion	gini, log loss
Min Impurity Decrease	0.0, 1e-8, 1e-10, 1e-12
Max Depth	150, 200, None

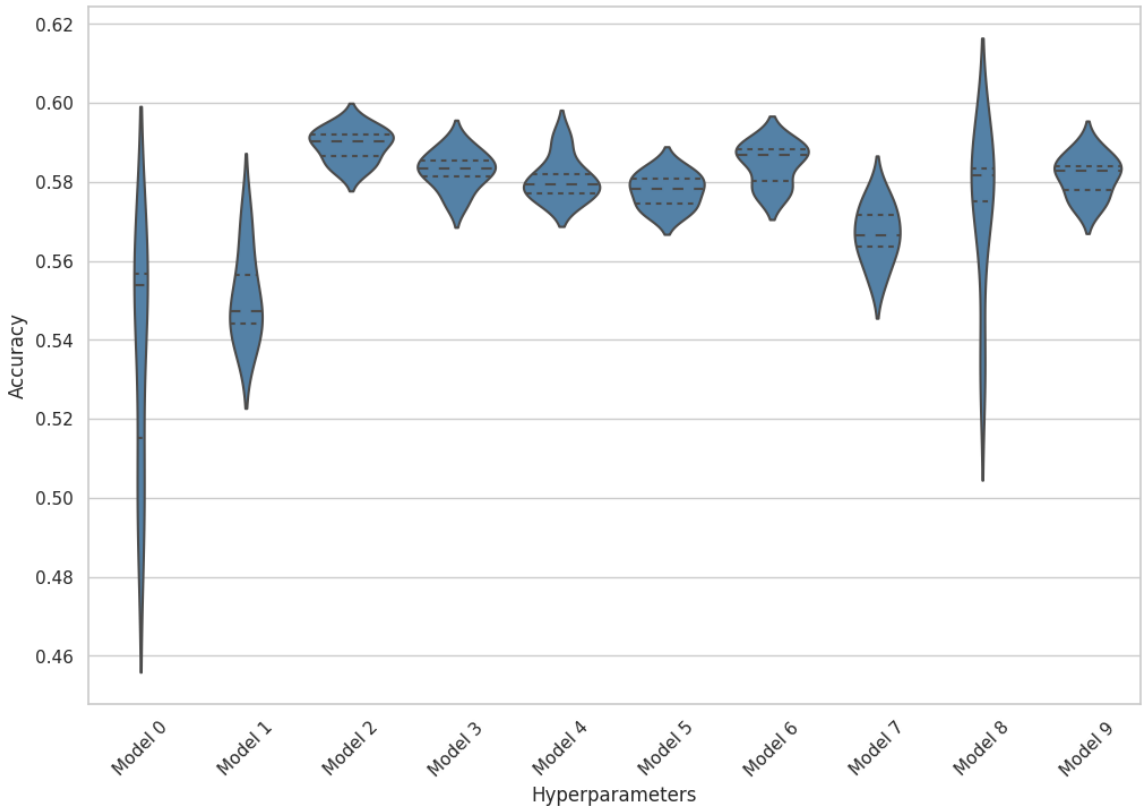
Experimental Results: Random Forest



- **Best Model**: Gini criterion, Max Depth of 150 and 150 estimators.
- **Accuracy**: Average accuracy of 58.89%.
- **Observation**: Improved performance with more estimators.

Hyperparameter	Value
Number of Estimators	50, 100, 150
Criterion	gini, log loss
Min Impurity Decrease	0.0, 1e-8, 1e-10, 1e-12
Max Depth	150, 200, None

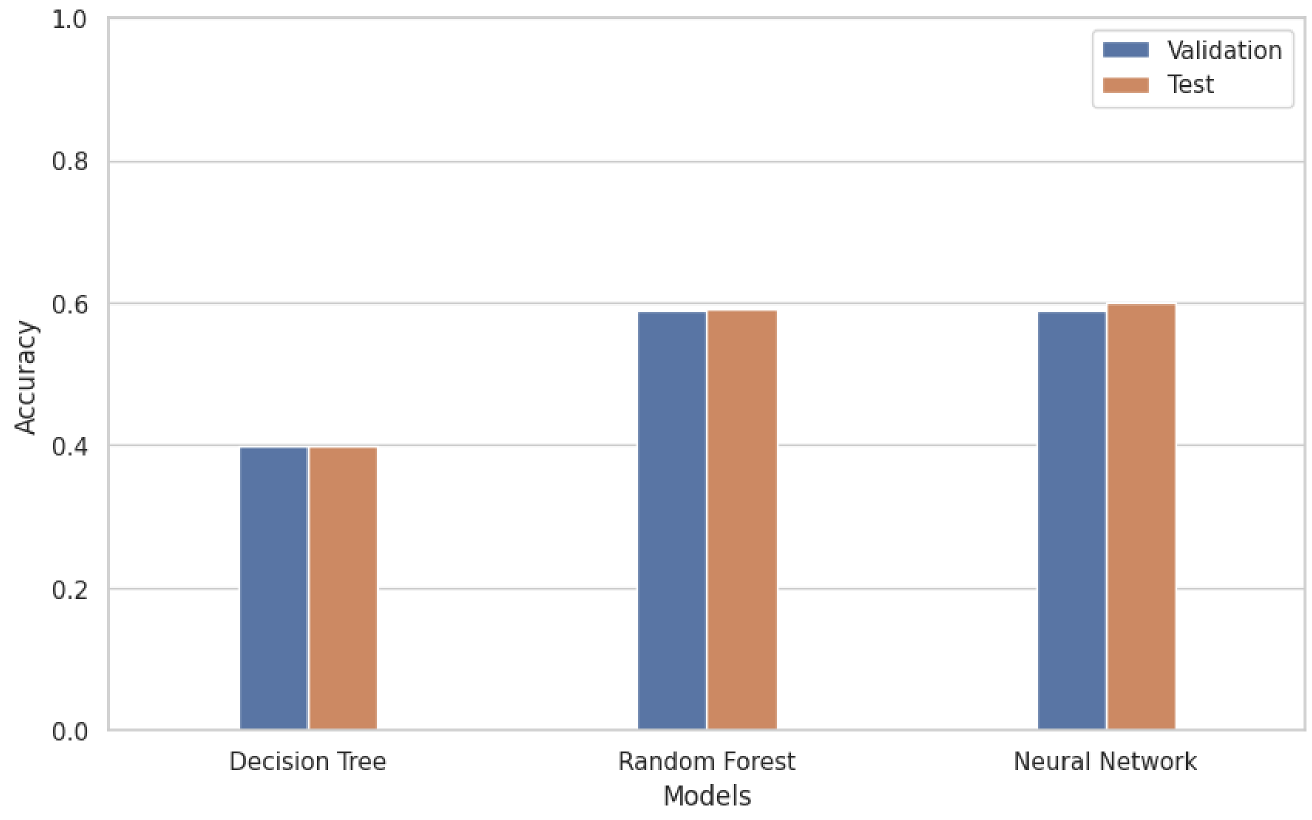
Experimental Results: Neural Network



- **Best Model**: Hidden size of 64, learning rate of 0.0005 and trained for 8 epoch.
- **Accuracy**: Average accuracy of 58.94%.
- **Observation**: Slightly better performance compared to Random Forest.

Hyperparameter	Value
Hidden Size	16, 32, 64
Epochs	6, 8, 10
Learning Rate	0.005, 0.001, 0.0005

Best Models Comparison



Tests Results

Neural Network: Best performance with 59.96% accuracy.

Random Forest: 59.25% accuracy.

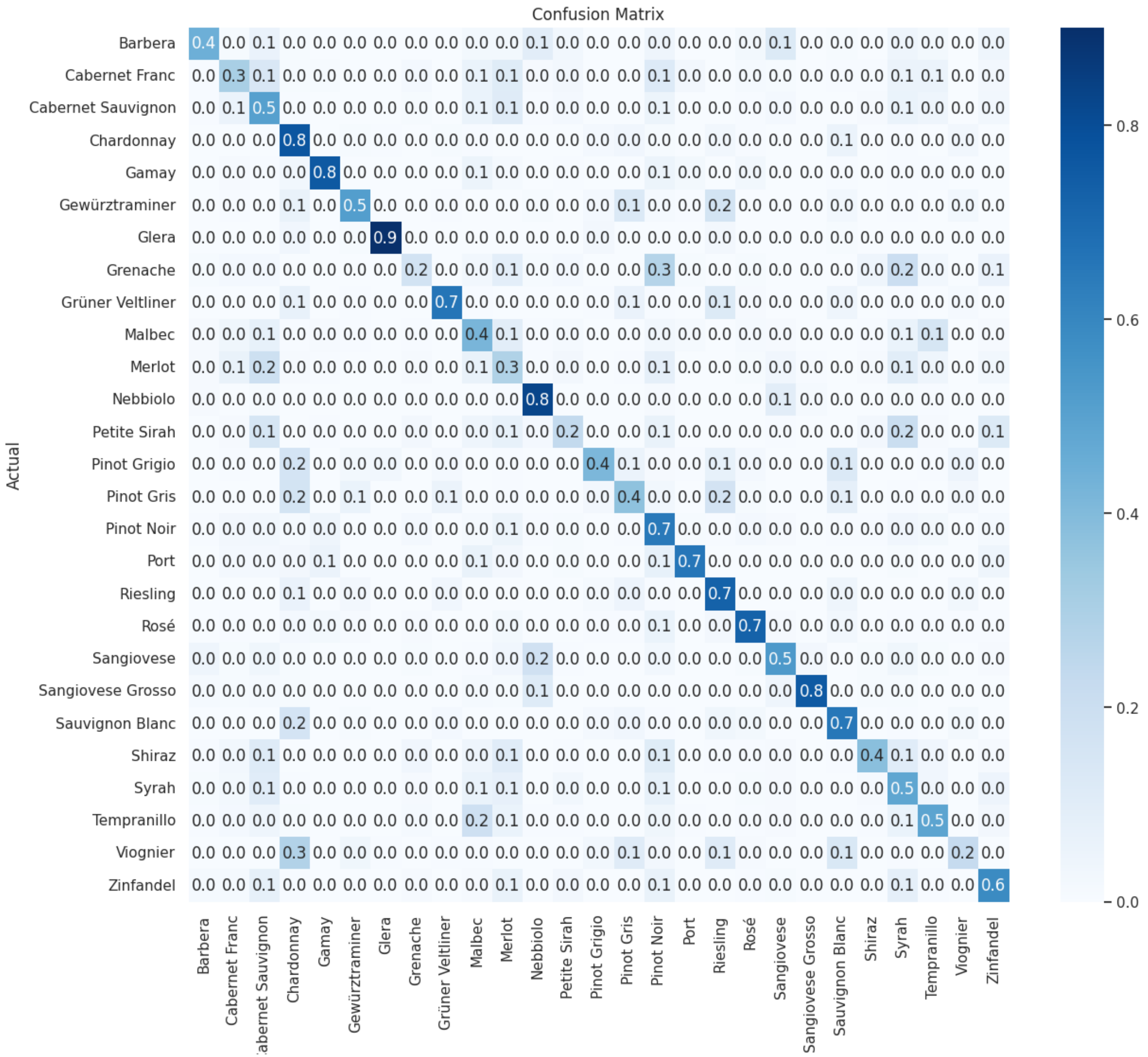
Decision Tree: 39.87% accuracy.

Wilcoxon Results

Decision Tree: 0.03125 p-value

Random Forest: 1.0 p-value

Neural Network Confusion Matrix



Conclusion and Future Works

Neural Networks excel in capturing *complex patterns* in text data, while simpler models like **Decision Trees** may *struggle* with intricate classification tasks.

Model Improvement: Further tuning of hyperparameters could boost performance.

Price Regression Analysis: Integrate a regression analysis for predicting wine prices based on reviews to provide more comprehensive insights.